[**https://msdn.microsoft.com/en-us/library/cc953fe1.aspx**](https://msdn.microsoft.com/en-us/library/cc953fe1.aspx)

**Fundamental Types (C++)**

**Visual Studio 2015**

[Other Versions](javascript:void(0))

https://i-msdn.sec.s-msft.com/Areas/Epx/Content/Images/ImageSprite.png?v=636107049706926704

Fundamental types in C++ are divided into three categories: integral, floating point, and void. Integral types are capable of handling whole numbers. Floating point types are capable of specifying values that may have fractional parts.

The [void](https://msdn.microsoft.com/en-us/library/fxky5d0w.aspx) type describes an empty set of values. No variable of type **void** can be specified — it is used primarily to declare functions that return no values or to declare generic pointers to untyped or arbitrarily typed data. Any expression can be explicitly converted or cast to type **void**. However, such expressions are restricted to the following uses:

* An expression statement. (See [Expressions](https://msdn.microsoft.com/en-us/library/625x66bt.aspx), for more information.)
* The left operand of the comma operator. (See [Comma Operator](https://msdn.microsoft.com/en-us/library/zs06xbxh.aspx) for more information.)
* The second or third operand of the conditional operator (**? :**). (See [Expressions with the Conditional Operator](https://msdn.microsoft.com/en-us/library/e4213hs1.aspx) for more information.)

The following table explains the restrictions on type sizes. These restrictions are independent of the Microsoft implementation.

**Fundamental Types of the C++ Language**

|  |  |  |
| --- | --- | --- |
| **Category** | **Type** | **Contents** |
| Integral | **char** | Type **char** is an integral type that usually contains members of the basic execution character set — By default, this is ASCII in Microsoft C++.  The C++ compiler treats variables of type **char**, **signed** **char**, and **unsigned** **char** as having different types. Variables of type **char** are promoted to **int** as if they are type **signed** **char** by default, unless the /J compilation option is used. In this case they are treated as type **unsigned** **char** and are promoted to **int** without sign extension. |
|  | **bool** | Type **bool** is an integral type that can have one of the two values **true** or **false**. Its size is unspecified. |
|  | **short** | Type **short** **int** (or simply **short**) is an integral type that is larger than or equal to the size of type **char**, and shorter than or equal to the size of type **int**.  Objects of type **short** can be declared as **signed** **short** or **unsigned short**. **Signed short** is a synonym for **short**. |
|  | **int** | Type **int** is an integral type that is larger than or equal to the size of type **short** **int**, and shorter than or equal to the size of type **long**.  Objects of type **int** can be declared as **signed** **int** or **unsigned** **int**. **Signed** **int** is a synonym for **int**. |
|  | **\_\_int8**, **\_\_int16**, **\_\_int32**, **\_\_int64** | Sized integer **\_\_int***n*, where *n* is the size, in bits, of the integer variable. **\_\_int8**, **\_\_int16**, **\_\_int32** and **\_\_int64** are Microsoft-specific keywords. Not all types are available on all architectures. **(\_\_int128** is not supported.) |
|  | **long** | Type **long** (or **long** **int**) is an integral type that is larger than or equal to the size of type **int**.  Objects of type **long** can be declared as **signed** **long** or **unsigned** **long**. **Signed** **long** is a synonym for **long**. |
|  | **long** **long** | Larger than an unsigned **long**.  Objects of type **long long** can be declared as **signed** **long long** or **unsigned** **long long**. **Signed** **long long** is a synonym for **long long**. |
|  | **wchar\_t**, **\_\_wchar\_t** | A variable of type **wchar\_t** designates a wide-character or multibyte character type. By default, **wchar\_t** is a native type, but you can use [/Zc:wchar\_t-](https://msdn.microsoft.com/en-us/library/dh8che7s.aspx) to make **wchar\_t** a typedef for **unsigned short**. The **\_\_wchar\_t** type is a Microsoft-specific synonym for the native **wchar\_t** type.  Use the L prefix before a character or string literal to designate the wide-character type. |
| Floating point | **float** | Type **float** is the smallest floating point type. |
|  | **double** | Type **double** is a floating point type that is larger than or equal to type **float**, but shorter than or equal to the size of type **long** **double**.  Microsoft specific: The representation of **long double** and **double** is identical. However, **long double** and **double** are separate types. |
|  | **long double** | Type **long** **double** is a floating point type that is larger than or equal to type **double**. |

Microsoft Specific

The following table lists the amount of storage required for fundamental types in Microsoft C++.

**Sizes of Fundamental Types**

|  |  |
| --- | --- |
| **Type** | **Size** |
| **bool**, **char**, **unsigned char**, **signed char**, **\_\_int8** | 1 byte |
| **\_\_int16**, **short**, **unsigned short**, **wchar\_t**, **\_\_wchar\_t** | 2 bytes |
| **float**, **\_\_int32**, **int**, **unsigned int**, **long**, **unsigned long** | 4 bytes |
| **double**, **\_\_int64**, **long double**, **long long** | 8 bytes |

END Microsoft Specific

See [Data Type Ranges](https://msdn.microsoft.com/en-us/library/s3f49ktz.aspx) for a summary of the range of values of each type.

For more information about type conversion, see [Standard Conversions](https://msdn.microsoft.com/en-us/library/aetzh118.aspx).

**See Also**

[Data Type Ranges](https://msdn.microsoft.com/en-us/library/s3f49ktz.aspx)